

УДК 681.3.6

ПОДСИСТЕМА ТРАНСФОРМАЦИИ БАЙЕСОВСКОЙ СЕТИ С ДЕТЕРМИНИРОВАННЫМИ СОСТОЯНИЯМИ

И.Н. Парасюк, Ф.В. Костукевич

Институт кибернетики им. В.М. Глушкова НАН Украины,
03680, Киев-187, проспект Академика Глушкова, 40.
Тел.: 526 6422, e-mail: ivpar1@i.com.ua

Рассмотрены новые более эффективные алгоритмы трансформации байесовских сетей в узловое дерево, полученные в результате модификации известных алгоритмов, состоящей во введении некоторой дополнительной структуры данных для хранения дуг сети, а также в выполнении процедуры предварительного упорядочения сепараторов. Конструктивно изложена функциональная модель и архитектура подсистемы, предназначенной для реализации процессов трансформации байесовских сетей с детерминированными состояниями.

The most effective algorithms for Bayesian network transformation into the junction tree which are received as a result of modification of the known algorithms, based on the ground of some additional structure data for preserving of arcs of network and also as a basis of doing procedure of previous sorting separators. Constructively given the functional model and architecture of computer additional system used for the realization of processes of transformation of Bayesian network with the determined states.

Введение

В последнее время для моделирования сложных систем с неопределенностями, как правило, недетерминированной природы широко применяется аппарат байесовских сетей (БС). Как известно [1], БС является нагруженным ациклическим орграфом со следующими соглашениями:

- каждый узел сети представляет собой событие, которому соответствует случайная величина (в контексте классической теории вероятностей [2]);
- дуга, направленная от узла X к узлу Y , означает, что X влияет на Y (узел X называется родительским (Parents) для узла Y);
- каждому узлу X_i сопоставляется некоторая функция условных вероятностей $P(X_i | \text{Parents}(X_i))$, с помощью которой количественно определяется влияние родительских узлов $\text{Parents}(X_i)$ на узел X_i ;
- граф не имеет ориентированных циклов (поэтому, является ориентированным ациклическим графом).

Для полноты изложения и однозначности понимания материала приведем также, придерживаясь [3], определения ещё некоторых ключевых понятий.

Пусть $G = (V, E)$ – БС. Обозначим V – множество, состоящее из n узлов данной сети, а E – множество всех дуг. Пусть каждому узлу этой сети соответствует единственная случайная величина из множества X_V .

Граф называется моральным (а процесс его создания – морализацией), когда он получен из БС после выполнения таких действий: если узел имеет несколько родительских узлов, тогда все они попарно соединяются между собой рёбрами, т.е. родительские узлы «женятся»; все дуги БС меняют на рёбра.

Кластером C_U , заданный над подмножеством $X_U \subseteq X_V$, называется подмножество $U \subseteq V$, в котором узлы сети объединены по определённом правилу, например, множество узлов БС, производящих кластер, образуют полный подграф графа G .

Кластерным деревом, заданным над множеством случайных величин X_V , называется неориентированное дерево, состоящее из кластеров, заданных над подмножествами множества X_V .

Сепаратором S_{AB} , который является нагрузкой для ребра между кластерами C_A и C_B , называется кластер, который задан пересечением подмножеств случайных величин $X_S = X_A \cap X_B$, т.е. $S_{AB} = C_A \cap C_B$.

Массой сепаратора S называют число узлов БС, входящих в его состав, а стоимостью сепаратора S называется сумма весов тех кластеров, которые соединяются данным сепаратором.

Кластерное дерево называется узловым или соединительным деревом, если для любых двух кластеров C_A и C_B , $C_A \neq C_B$, выполняются условия: $C_A \not\subset C_B$ и $C_B \not\subset C_A$, а также на пути между любой парой кластеров C_A и C_B объединение всех сепараторов содержит пересечение $C_A \cap C_B$.

Пусть A, B, C – три разных подмножества множества узлов V БС G . Говорят, что для морального графа G^m , построенного на основе БС G , можно выполнить декомпозицию (декомпонировать), если $V = A \cup B \cup C$ и выполняются следующие утверждения: C – сепаратор для кластеров A и B ; C – полный подграф графа G^m .

Граф называется хордальным, если в нём каждый цикл, длиной более 3, имеет хорду. Поскольку любой простой цикл в хордальном графе является треугольником, то такой граф ещё называют триангулированным, а процесс его получения – триангуляцией [4].

Существенную роль в развитии и применении БС с детерминированными состояниями для эффективного моделирования сложных процессов и систем сыграл алгоритм передачи сообщений (иначе алгоритм распространения доверия), предложенный Джуди Перлом [5]. Одной из центральных проблем, возникающих при применении алгоритма передачи сообщений на основе БС, является проблема трансформации ориентированного ациклического графа в кластерное (узловое) дерево, позволяющее ускорить вероятностный вывод и при этом рационально использовать память компьютера. Доказано [3], что для любой БС существует узловое дерево тогда и только тогда, когда моральный граф данной сети можно декомпонировать на связное множество полных подграфов, а декомпонировать моральный граф можно тогда и только тогда, когда он является хордальным или преобразован в хордальный граф. Из этого следует, что трансформацию БС в кластерное дерево можно представить последовательностью преобразований [3, 6]: выполнить морализацию БС и создать моральный граф; выполнить триангуляцию морального графа и создать хордальный граф; выполнить декомпозицию хордального графа на полные подграфы (кластеры); создать множество сепараторов, соединяющих кластеры в узловое дерево.

Наиболее принципиальный и трудоемкий шаг в процессе трансформации БС – это триангуляция морального графа, являющаяся NP-трудной задачей относительно произвольной БС. Нетривиальным является также алгоритм преобразования хордального графа в кластерное дерево, поскольку требуется проведение анализа всевозможных связей между вершинами кластерного дерева.

Авторами предложена модификация известных алгоритмов, состоящая во введении некоторой дополнительной структуры данных для хранения ребер сети, а также в выполнении процедуры предварительного упорядочения сепараторов. Используя такую модификацию, удалось построить алгоритмы трансформации БС, которые на порядок быстрее известных [6]. В качестве полигона для проведения вычислительных экспериментов по реализации трансформаций разработана соответствующая инструментальная среда в виде подсистемы трансформации БС (далее ПТБС). Подсистема реализует трансформации БС на основе разработанных, модифицированных алгоритмов, предназначенных для построения хордального графа и узлового дерева, а также выполняет визуализацию процесса трансформации БС. В настоящем докладе излагаются наиболее принципиальные вопросы построения данной подсистемы.

Функциональная модель и архитектура ПТБС

Основными функциями ПТБС, реализующими процессы трансформации, являются: создание экземпляра БС на основе встроенного в ПТДС генератора БС или на основе XML-документа, содержащего информацию о структуре БС; морализация графа, а также визуализация исходной БС и морального графа; триангуляция графа, а также визуализация исходной БС и хордального графа; построение узлового дерева, а также визуализация исходной БС и узлового дерева; сохранение БС и результата трансформации в графическом формате (формат bmp).

Доступ к функциям реализован в виде соответствующих команд меню графического интерфейса ПТБС, что предоставляет возможность пользователю неоднократно выполнять преобразования любой БС, а также в удобном для себя темпе проанализировать полученные результаты и при необходимости сохранить их в графическом формате. На рис. 1 показан фрагмент XML-схемы, содержащий информацию общего характера о БС (наименование сети, количество узлов), а на рис. 2 – информацию о каждом узле.

```
<xs:element name="network" type="networkType"/>

<xs:complexType name="networkType">
  <xs:sequence>
    <xs:element name="net_name" type="xs:string"/>
    <xs:element name="count_node" type="xs:long"/>
    <xs:element name="node" type="nodeType"/>
  </xs:sequence>
</xs:complexType>
```

Рис. 1. Фрагмент XML-схемы для определения общего характера БС

```

<xs:complexType name="nodeType">
  <xs:sequence>
    <xs:element name="node_name" type="xs:string"/>
    <xs:element name="node_level" type="xs:long"/>
    <xs:element name="node_parents" type="parentsType"/>
    <xs:element name="node_size" type="xs:long"/>
    <xs:element name="node_states" type="node_statesType"/>
  </xs:sequence>
</xs:complexType>
...
</xs:schema>

```

Рис. 2. Фрагмент XML-схемы для определения структуры БС

Генератор БС работает в двух режимах: пользователь вводит количество узлов сети, количество состояний и количество уровней, необходимых для визуализации сети; пользователь вводит количество узлов сети, а также параметры для каждого узла (имя, родительские узлы, количество состояний, уровень для визуализации узла).

После визуализации входной информации, т.е. БС, пользователь с помощью графического интерфейса имеет возможность выполнить последовательные преобразования экземпляра БС в узловое дерево (согласно алгоритму трансформации). В системе использовались авторские модификации алгоритмов триангуляции и построения узлового дерева [2], позволяющие уменьшить время выполнения соответствующих алгоритмов (табл. 1: n – количество узлов в БС, k – максимальное для каждого узла количество смежных узлов).

Таблица 1. Оценки временной сложности некоторых алгоритмов трансформации БС

Версия алгоритма	Алгоритм триангуляции	Алгоритм узлового дерева
Оригинальная версия	$T(nk)=O(k^2n^2+n^2+k^2n+n)$	$T(n)=O(n^4-n^3+n^2+n)$
Модифицированная версия	$T(nk)=O(n^2+k^2n+kn+n)$	$T(n)=O(n^2+n\log(n)+n)$

На рис. 3 показана реализация архитектуры ПТБС, выполненной на языке программирования C++ (прямоугольник изображает модуль ПТБС, круг – интерфейс модуля, сплошная линия обозначает, что модуль предоставляет интерфейс другому модулю, пунктирная – используется интерфейс соответствующего модуля (согласно нотации UML)).

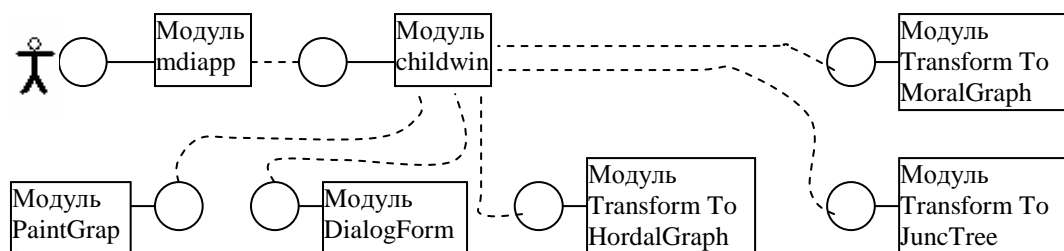


Рис. 3. Модель реализации архитектуры ПТБС в виде взаимодействующих модулей

Модуль mdiapp – создаёт родительское окно системы и предоставляет к нему графический интерфейс. Через интерфейс родительского окна пользователь управляет созданием экземпляра БС, а также дочерними окнами, каждое из которых содержит свой экземпляр БС. Другими словами, пользователь может одновременно анализировать трансформации нескольких БС. Модуль childwin – создаёт дочернее окно системы и предоставляет к нему графический интерфейс, а также интерфейс для вызова методов поэтапной трансформации БС. В дочернем окне системы отображается исходная БС и визуализируется результат очередного этапа трансформации. Модуль DialogForm – предоставляет графический интерфейс для настройки и запуска генератора БС. Модуль TransformToMoralGraph – предоставляет программный интерфейс для преобразования БС в моральный граф. Входными данными для алгоритма является БС, а результатом – моральный граф. Этот модуль содержит алгоритм морализации БС, состоящий из шагов.

1. Для каждого узла БС:

найти все пары родительских узлов и соединить их ребрами.

2. Все дуги БС заменить рёбрами.

Полученные после морализации БС группы узлов (узел + все его родительские узлы) становятся основой для создания кластеров на следующем этапе трансформации.

Модуль TransformToHordalGraph – предоставляет программный интерфейс для преобразования морального графа в хордальный, т.е. содержит алгоритм триангуляции морального графа (используется дополнительная структура H для хранения предварительно построенных списков хордальных рёбер) и создания кластеров для узлового дерева. Алгоритм триангуляции состоит из шагов.

1. Для каждой вершины v морального графа:

создать подмножество h_v , состоящее из трёх частей – список хордальных рёбер, их количество и вес

кластера; $H = H \cup h_v$.

2. Установить начальное значение счётчика вершин $i = n$.

3. Пока существуют нумерованные вершины:

среди нумерованных вершин v морального графа найти такое подмножество $h_v \subseteq H$, количество хордальных рёбер которого было б минимальным; если таких подмножеств h_v несколько, тогда среди них выбрать то, которое порождает кластер наименьшего веса;

добавить в моральный граф хордальные рёбра из подмножества h_v . Сохранить кластер C_v : $C = C \cup C_v$;

удалить вершину v_i вместе с инцидентными ей рёбрами в моральном графе, а также из всех подмножеств h_u , где u – смежная с v вершина;

уменьшить счётчик i на 1.

Модуль TransformToJuncTree – предоставляет программный интерфейс для построения узлового дерева из кластеров, полученных в результате триангуляции. Основная идея авторского варианта алгоритма – это упорядочивание множества сепараторов перед циклом, в котором строится узловое дерево. Упорядочивание такого множества сводится к сравнению целых чисел, которые обозначают массы и стоимости сепараторов. Поэтому для сортировки сепараторов можно выбрать любой алгоритм сортировки, время выполнения которого $n \log(n)$ [7]. Схема модифицированного алгоритма узлового дерева.

1. Создать пустое множество S , в котором сохраняются все сепараторы, и S' , в котором сохраняются сепараторы узлового дерева.

2. Для каждой пары кластеров C_A та C_B :

создать сепаратор $S_{AB} = C_A \cap C_B$;

добавить сепаратор к множеству $S = S \cup S_{AB}$.

3. Упорядочить множество сепараторов по убыванию масс. Если массы одинаковые, тогда упорядочить по возрастанию стоимости сепараторов.

4. Пока не все кластеры соединены в дерево, выполнять:

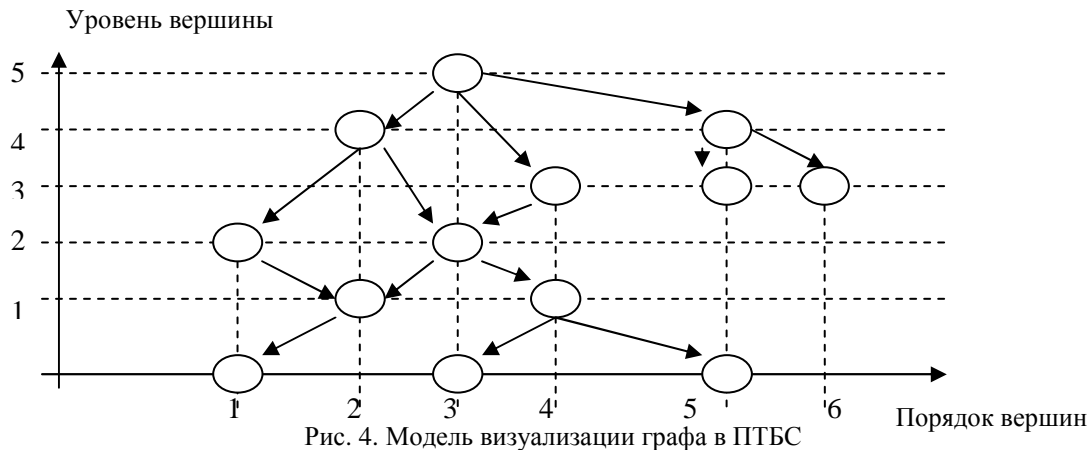
выбрать из множества сепараторов S очередной сепаратор;

если два кластера C_A и C_B , соединённые сепаратором S_{AB} , принадлежат к разным деревьям, тогда объединить эти деревья в одно общее дерево, соединив ребром кластеры C_A и C_B ;

если кластеры C_A и C_B были соединены новым ребром, тогда выбранный сепаратор S_{AB} из множества S добавить во множество сепараторов узлового дерева S' , т.е. $S' = S' \cup S_{AB}$.

После выполнения алгоритма узлового дерева в кластерах объединяются узлы исходной БС. Поскольку, согласно с определением БС, каждому узлу X_i ставится в соответствие функция условных вероятностей $P(X_i | \text{Parents}(X_i))$, с помощью которой количественно определяется влияние родительских узлов $\text{Parents}(X_i)$ на узел X_i , то аналогичные функции определяются для каждого кластера и сепаратора отдельной процедурой в модуле TransformToJuncTree. Для каждого кластера функция определяется на множестве узлов БС, входящих в кластер, и вычисляется с помощью перемножения всех функций условных вероятностей $P(X_i | \text{Parents}(X_i))$, заданных для каждого узла X_i . Для каждого сепаратора функция определяется на множестве узлов БС, входящих в сепаратор, и тождественно приравнивается к единице. Произведение функций, полученных для кластеров и сепараторов узлового дерева, согласно [3], определяет функцию общего распределения вероятностей, заданную над множеством узлов первоначальной БС. Таким образом, трансформация БС позволяет получить графическую интерпретацию общего распределения вероятностей, разложенного на множители, в виде кластеров узлового дерева. Кроме того, согласно с алгоритмом распространения доверия [5], над каждой функцией, определённой над одним кластером, можно выполнить вычисления независимо от действий выполняемых над функциями, определёнными над другими кластерами. Данный факт позволяет уменьшить сложность вычислений и рационально использовать память компьютера для сохранения основных и промежуточных результатов.

Модуль PaintGraph – предоставляет программный интерфейс для визуализации БС, морального графа, хордального графа, узлового дерева. Модель, использованная в ПТБС для визуализации, показана на рис. 2. В основе модели использована идея изображения вершин графа на соответствующих уровнях. Уровни для вершин задаются пользователем или программно (в зависимости от способа создания графа). Размещение вершин на уровне выполняется на основе алгоритма, который анализирует связи между вершинами соседних уровней и, на основе анализа уменьшает количество пересечений рёбер. В результате применения такого алгоритма каждая вершина на плоскости однозначно определяется двумя координатами: по вертикали – уровень вершины, по горизонтали – порядок вершины на уровне.



Информационные структуры, которые использует ПТБС для обработки графов и их элементов (вершин, рёбер), показаны на рис. 5.

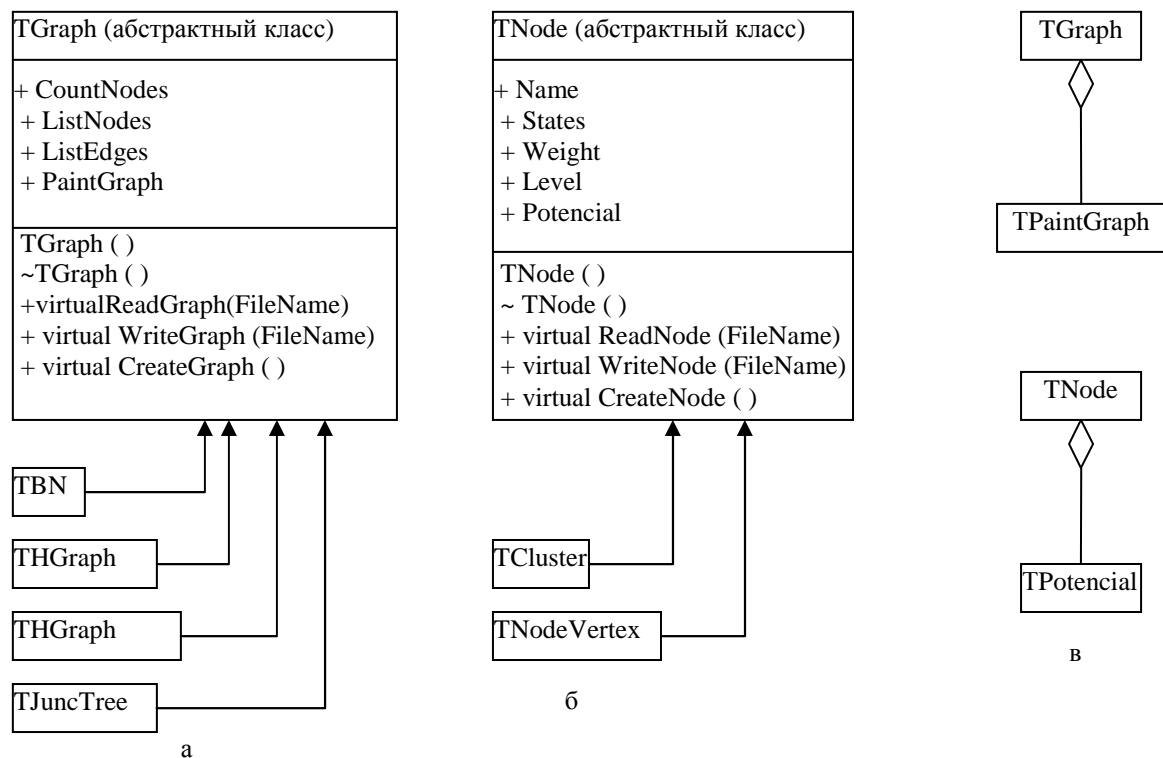


Рис. 5. Диаграммы классов для ПТБС

Базовым классом для графовых структур является абстрактный класс TGraph, содержащий атрибуты, описывающие весь граф, виртуальные методы, определяемые индивидуально в каждом дочернем классе TBN, TMGraph, THGraph, TJuncTree (рис. 5, а). Класс TBN предназначен для хранения информации о структуре БС, необходимой для визуализации графа и последующей его трансформации в моральный граф. Класс TMGraph предназначен для хранения информации о структуре морального графа, необходимой для его визуализации и последующей трансформации в хордальный граф. Класс THGraph предназначен для хранения информации о структуре хордального графа, необходимой для его визуализации, а также информацию о кластерах. Класс TJuncTree предназначен для хранения информации о структуре узлового дерева, необходимой для его визуализации, а также информацию о кластерах, сепараторах и распределениях вероятностей над каждым кластером и сепаратором.

На рис. 5, б представлена иерархия классов TNode, TNodeVertex, TCluster, позволяющая обрабатывать информацию о вершинах разных графов, используемых в ПТБС. Класс TNode является абстрактным классом, содержащим атрибуты, которые являются общими для вершин графов. Класс TNodeVertex предназначен для хранения и обработки информации, связанной с узлами БС и вершинами морального и хордального графов. Класс TCluster предназначен для хранения и обработки информации, связанной с кластерами и сепараторами

узлового дерева. Класс TGraph содержит атрибут PaintGraph, который является указателем на класс TPaintGraph, т.е. TGraph агрегирует класс TPaintGraph (рис. 5, в) – вверх). Класс TPaintGraph необходимый для хранения функций-членов, использующиеся для визуализации графов. Базовый граф TNode содержит указатель на класс TPotencial, описывающий функцию распределения вероятностей для соответствующей вершины графа, т.е. класс TNode агрегирует класс TPotencial (рис. 5, в) – вниз). Классы TBN, TMGraph, TNGraph агрегируют TNodeVertex, а класс TJuncTree – класс TCluster.

Построенная ПТБС тестировалась на модельных БС. Результаты тестирования совпадают с расчётными, выполненными на основе [1, 5]. В качестве примера показаны на рис. 6, а) – результаты трансформации БС. На рис. 6, б) показан результат морализации БС (пунктиром показаны добавленные моральные рёбра, соединяющие родительские вершины с общей дочерней вершиной). На рис. 6, в) показан результат триангуляции морального графа, добавлены рёбра (обозначены пунктиром с точкой), позволяющие объединить вершины графа в кластеры. На рис. 6, г) показано узловое дерево, построенное на основе хордального графа (рис. 6, в).

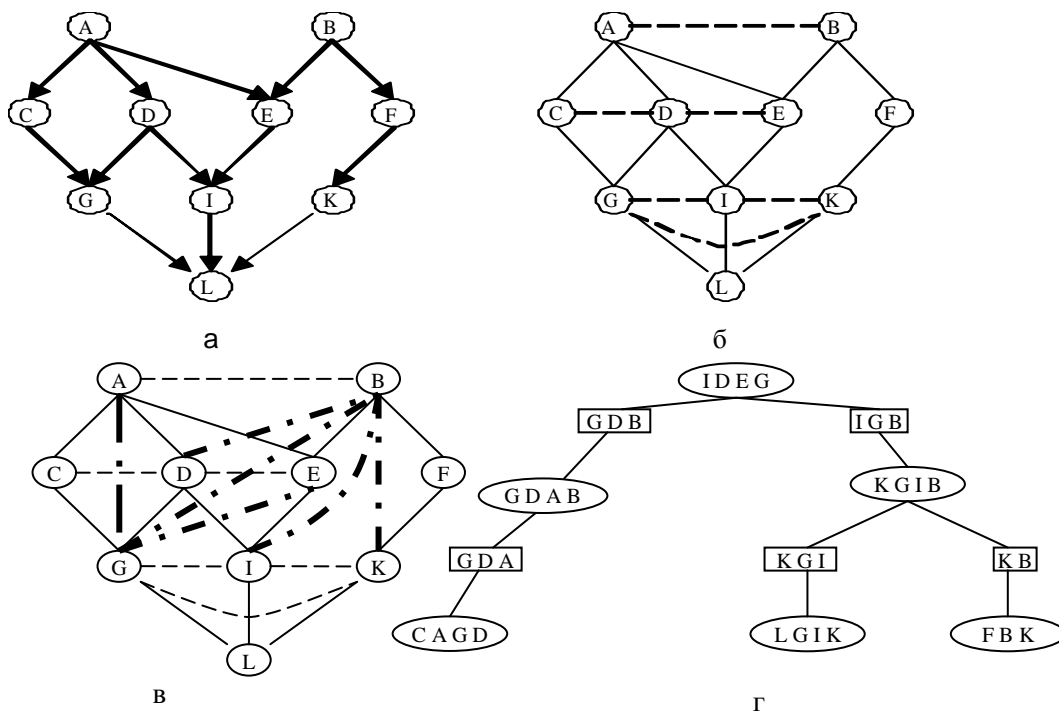


Рис. 6. Последовательное преобразование БС (а) в узловое дерево (г)

Заклучение

Следует отметить, что разработанная авторами компьютерная подсистема ПТБС предоставляет унифицированные механизмы трансформации БС независимо от природы состояний ее переменных. В частности, данные механизмы могут быть использованы в качестве составных компонент информационных технологий для моделирования и исследования сложных процессов и систем на основании БС с недетерминированными состояниями, например размытыми.

1. Рассел С., Норвиг П. Искусственный интеллект: современный подход, 2-е изд.: Пер. с англ. – М.: Издательский дом «Вильямс», 2006. – 1408 с.
2. Пугачёв В.С. Теория вероятностей и математическая статистика: Учеб. пособие. – 2-е изд., испр. и дополн. – ФИЗМАТЛИТ, 2002. – 496 с.
3. Cowell R.G., Dawid A.P., Spiegelhalter D.J., Lauritzen S.L. Probabilistic Networks and Expert Systems. – Springer-Verlag, New York, Inc, 1999. – 321 p.
4. Касьянов В.Н., Евстигнеев В.А. Графы в программировании: обработка, визуализация и применение. – СПб.: БХВ-Петербург, 2003. – 1104 с.
5. Pearl J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. – Morgan Kaufmann, San Mateo, CA, 1991. – 552 p.
6. Парасюк И.Н., Костукевич Ф.В. Методы трансформации байесовской сети для построения узлового дерева и их модификация // Компьютерная математика. – 2008. – №1. С. 58 – 68.
7. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ, 2-е изд.: Пер. с англ., под ред. А. Шеня. – М.: МЦНМО: БИНОМ, Лаборатория знаний, 2004. – 960 с.